

Bounding Ambiguity: Experiences with an Image Annotation System

Margaret Warren¹, Patrick Hayes²

¹ Metadata Authoring Systems

² Florida IHMC,

Abstract. This paper reports on a web interface for creating and editing rich metadata descriptions for images using RDF triples. We discuss the roles of ambiguity, disagreement and subjectivity in knowledge formation arising from an ongoing experiment in which users create semantic annotation of images, and we discuss the ways these elements have influenced the design of the system.

1 Background

ImageSnippets (<http://www.imagesnippets.com>) is a web interface designed for the construction of machine-computable image descriptions by lay users. Prompted by image cues, image annotators and/or subject matter experts can disambiguate entities from public corpora or define new terms through the interface. The subsequent captured knowledge is then stored in triple based graphs for inference, findability and reuse by semantically aware processes. The output formats conform to W3C linked data standards: (RDFa, JSON-LD) and utilize terms from the large and growing web-based concept datasets including DBpedia, YAGO and the Art & Architecture Thesaurus, without requiring users to be aware of this machinery.

The system has a number of features for capturing the highly subjective things people might say about images, but it also allows them to disambiguate their meanings in a suitably precise way, thereby providing a process for formalizing intuitive and expert knowledge. In our research with the system, we have found disagreements about meanings to be a centrally important methodological tool to create more useful and intuitive conceptual distinctions and that ambiguity is unavoidable and often useful, and should be considered something to be controlled and bounded appropriately, rather than eliminated; and that shared subjectivity is more important than objective truth or correctness when using semantic markup for inference and retrieval. Our work originally began as an exploration of how to bring together precise machine-readable descriptions in Semantic Web notations such as RDF and OWL with the informal language of how people actually describe images, especially (but not exclusively) the language of artists and deliberately considered the artists themselves as the subject matter experts on their own work (Eskridge et al, 2006, Warren & Hayes 2007, 2010). But this early knowledge representation work also had a secondary goal: that we could use our work to design a system with a (hopefully) intuitive interface that would capture this highly subjective, personalized

and informal knowledge in such a way that the resulting precise and disambiguated metadata could improve findability and then be distributed on the web with as much readability, interoperability and persistence as possible.

So the ImageSnippets system was designed first and foremost as a research tool to observe how users or image annotators would interact with the system, but we also considered that the metadata created through this research could also be accessed by any semantically aware processes and search engines and further, that the data could be properly attributed with provenance and publishing authority.

At the core of the ImageSnippets system is an intentionally small, lightweight ontology which provides a basic vocabulary of properties used to relate an image - the subject of the descriptions (or a region in the image) - to object values as entities which are then selected from public data sets and which, in other image annotation systems, would normally be referred to as keywords or tags. The annotators are given these properties as intuitive guidelines and encouraged to use terms from LIO first, but with some training, they can create new properties as well. The ability for users to create new properties is a highly useful way for the system to capture and analyze subtle distinctions and use these distinctions to build both new, domain specific ontologies and to extend LIO to allow annotators to talk, not just about images, but about the things that are seen in the images. LIO has evolved over many thousands of hours of use and testing, and the history of this evolution provides some suggestive lessons in the use of contradiction and ambiguity to discover intuitive concepts.

The design of the main ImageSnippets interface by necessity had to be concerned with capturing highly subjective and ambiguous sentiments from the naturally expressed ways people describe images, but incorporated in the design was also the creation of the core terms which make up LIO and this process was, in itself an iterative exercise in using something we call *bounded ambiguity*.

Outside of mathematics and some related pursuits, the meanings of words or symbols are always ultimately determined by the way that those words are actually used to pragmatically convey content, rather than by exact formalizable definitions. Even though Web protocols give IRIs an unprecedented degree of global exactness when used to retrieve content, this does not apply to their use as names in the Semantic Web, where their meanings are just as socially defined as words in natural languages. Widely cited calls to remove all ambiguity from IRIs (such as <https://www.w3.org/wiki/GoodURIs>) are doomed to failure.

Nevertheless, while ambiguity cannot be eliminated, it can be bounded. Published OWL ontologies and thoroughly documented public catalogs of IRI meanings such as DBpedia can both give IRIs a tighter, more exactly restricted meaning than is commonly found in natural-language words, and sometimes can adequately capture useful meaning distinctions which have not (yet) been adopted in normal English. Much of the art in designing useful Web data seems to involve locating the most useful bounds on ambiguity and we believe that bounded ambiguity has yielded some highly useful and encouraging search results.

Naturalness of use, and plausibility of the resulting descriptions, have been central to the goal of our project, which aimed to take image description to a level impossible to reach by the use of simple 'keywords' or 'tags'. Image tags or keywords cannot overcome the sometimes extreme lexical ambiguity of English words in isolation, and cannot record how the indicated idea is related to the image. It should be noted here

that there are many existing annotation systems and crowdsourcing efforts that employ both complex schemas and keyword disambiguation as well as the ability to group keywords into more useful machine processable conventions; however the subject of our research has always been to push the edges of keyword meaning in relationship to the image.

Therefore, ours is a very different kind of activity from conventional ontology engineering. It is mostly performed by people experienced in the use of the software but with no philosophical or technical background, and in some cases with a high-school level of education. It rarely strays into ‘upper-level’ decisions about how things are to be classified, beyond a very basic and shallow taxonomy of classes of things. It is driven by the immediate needs of describing things seen in images, and failure to say something is not an acceptable option, so the subject matter is open-ended. Naturalness is essential, but descriptions cannot go beyond the restricted grammatical patterns of RDF linked triples, so concepts must sometimes be invented to express complex ideas. When this is done, the results are subjected to a Darwinian process of selection: if a new idea – usually a new relation name – is found to be useful in other, subsequent, annotations, it is retained, and this re-use in itself provides a growing body of evidence illustrating its actual meaning. We believe that, in this way, a slowly growing corpus of useful concepts - mostly RDF properties - are being accumulated which seem to allow quite rich expressiveness within the confines of the simple RDF triple graph syntax.

One important idea in our methodology has been that of the information recording point, where a user has a clear thought about an image and should be able to express this naturally, in an annotation which is detailed and structured enough to transmit that intended meaning to other users downstream. The information recording point is the moment that the annotator has something to say and is ready to say it. Our aim is to provide a tool that can be used at that point, without the annotator being required to see or think about anything beyond ideas expressed in English words and a readable rendering of their content. (The interface provides this as a grammatically primitive but readable ‘pidgin’ pseudo-English.) Coupled with this is an idea borrowed from software training usually in corporate cultures, that of ‘just-in-time learning’. In a way, it could be said that we are giving annotators ‘just enough’ structure and just enough guidance in the interface to choose from one of the core concepts at the very moment the annotator is deciding what they want to say. Most users of the system can be trained in the basic construction of triples using the core LIO concepts in just a few minutes and after they have gained sufficient experience in the basics, they can easily then be trained to extend the triples to allow even greater expressivity.

The inherent ambiguity of the core LIO vocabulary seems to be part of the reason for its success. If we had imposed very ‘strict’ meanings on these relations, making finer-grained distinctions based on very exact meanings, it would of necessity have been several orders of magnitude larger, greatly increasing the cost of development but also the cost of use, since users would be required to make very precise distinctions at the point of use, with low tolerance for error.

It is difficult to talk about the design of LIO without also discussing some of the interface design decisions, because the vocabulary was deliberately allowed to grow somewhat organically around a combination of user workflow and some key philosophical decisions as starting points. So we first created the editor to enable the

triples to be written and almost immediately designed a complementary search and sort function to test the effectiveness of the properties. Additionally, as part of building the triples, users were provided with a look-up for finding and disambiguating the object values of the triples from public datasets. Choosing entities from sources such DBpedia, Yago, The Art & Architecture Thesaurus is also a highly subjective process which itself influences the choices made for properties.

To date, the basic LIO ontology consists of eleven properties (including a small number used to classify images into categories, such as being a photograph or a pencil drawing, and placing them in collections) have been found to be sufficient, together with the concepts from DBpedia and other online corpora, to create rich and useful descriptions of a wide variety of images.

2 Building LIO

Almost every aspect of the LIO ontology has its origin in a *disagreement*, either between ourselves or between us and some of our users. Sometimes these were resolved by careful disambiguation; but more often, by realizing that we were all using words differently, leading to the creation of a new concept name, whose true meaning was often decided by the patterns of its actual use. Some examples follow.

2.1 Images vs Works

The root subject IRI of all our descriptions identifies the digital image shown to the user as a thumbnail, but exactly what this IRI is supposed to denote in our descriptions is not entirely obvious, and was heavily debated. For a digital photograph the root subject is simply the image itself, but for a photograph of a painting in a museum catalog it is natural to say that ‘the image’ refers to the painting, and in some cases –for example, a cropped image of a sculpture in a museum catalog – it may even be a physical object. In general, we use the terminology of a *work*, as in ‘work of art’ and the subject of our descriptions is always the work.

But this introduces an ambiguity: is our subject IRI referring to the image itself or to a work seen in the image? At an early stage of the design of our interface we resolved this ambiguity by providing a carefully described set of alternatives, but this proved to be unworkable, as it required users to carefully make artificial-seeming distinctions, and to create awkward descriptions. If one delves into these distinctions deeply, one quickly descends into an intellectual quicksand. Who exactly is the creator of what you see in a photo of graffiti? Most people would say the photographer, unless it is something recognizable by an artist like Banksy. In our experience, the way an image is cropped seems to have some bearing on who might be said to be the creator. In some cases, it is the Contact Volume Editor that checks all the pdfs. In such cases, the authors are not involved in the checking phase.

We simply assume that the subject of our descriptions is a work, without recording explicitly whether this means the image itself or something it illustrates. (Intuitive guidelines are that a cropped, sharp image of a single object shown in an idealized

studio-lighting format is a picture of the work, while a simple photographic image of a landscape is the work. These distinctions are not recorded explicitly, so the image IRI might refer to itself or to a work it illustrates. This is of course the classical philosophical confusion of use versus mention. This is an ambiguity that we have decided to embrace rather than disambiguate, which has not turned out to be a problem in practice. Users seem to follow the required discipline intuitively without needing to be trained or corrected. We think that there are probably good reasons why some famous philosophical confusions are widespread in human affairs, even though they have been noted by scholars for thousands of years. The time or effort they save, compared to more accurate but more pedantic descriptions, is enormous, while the ambiguity they embody is easily resolved in practice. In our case, the use of multiple sources of information in descriptions, including the use of a ‘creator’ field, or the Dublin Core *dc:creator* property, or both, seems adequate.

2.2 Depiction vs Showing

The most basic LIO property is *depicts*. We debated reusing the depiction properties used in FOAF (Brickley & Miller 2010). But in spite of the widely recognized best-practice utility of re-use, we decided to introduce and use *lio:depicts* as a new property on the grounds that almost all FOAF uses will be images of people, whereas LIO is intended to apply to a much wider range of images. While nothing in the published FOAF vocabulary description requires this restriction to people, it seems clear that in actual use, the domain of foaf:depiction is people rather than owl:Thing; and we believe that use determines actual meaning.

Real images often show many entities that are incidental to the main subject (if there is one). Are they all depicted? A disagreement about pictures which ‘accidentally’ reveal a celebrity, while being aimed at other subjects, led us to introduce a distinction between depicting and merely showing. We noted that other systems exist which introduce a concept of weighting how central each thing depicted is in a scene, so we introduced a property *lio:shows* to be used for noting things in images which are in a sense incidental or not the ‘main subject’. The distinction between *lio:depicts* and *lio:shows* is vague and underdetermined, but it seems natural, and there is a high degree of agreement in the use of these relations by different users in all the cases we have tested. Also, people use these terms with very little training and they use them rapidly, in much the same amount of time as it would take to create bare keyword tags. Figure 1 illustrates some examples of how the distinction works. Subsequently, all of the other LIO properties seem to be split away from ‘depiction’



where a visual ambiguity needs to be resolved.

Figure 1: *Depicts, Shows, UsesPictorially*

2.3 Location vs. Setting

Early uses of a draft of the ontology revealed a form of use that we had not anticipated. It can be illustrated by the example of a photograph of some newly prepared food on a table. The photographer had added the description

lio:shows dbpedia:Pensacola,_Florida

on the grounds, when asked, that the photo had been taken in Pensacola. Clearly he was not using our property in the way we had intended. When we explained his error, his response was to say that he felt the ‘setting’ of the photograph was important.

It is worth noting that the issue of location in image annotation is in and of itself highly ambiguous and probably hotly contested in many circles. Early on, we had added a static location field (re-using an Adobe XMP location that can often be found embedded in image data) to attempt to capture at least one version of location initially; but there are at least 4 notions of location: the location of the camera or input device that captured the digital image, the location where a work might have been created, the location of a scene in the work and the location of where the work may currently be located. In all of these cases, attempts have been made to classify the distinction between scene location versus image location — and even the names of the location fields themselves become ambiguous and tend to change over time.

So we created a new property, *lio:hasSetting*, to relate an image to the broader place where it was created while still providing in other places in the interface, a way for annotators to deal with camera or scene locations. This judicious use of bounded ambiguity was rapidly overtaken by our users, who quickly extended its use further than we had imagined. They included events (the 1939 NY World’s Fair), periods of time (the 17th century, when a portrait was painted) and even such things as ‘kitchen’, ‘wedding’, and even a person – for the setting of a tattoo. While these uses were not what we had in mind, they seem to work, and indeed we now find them natural. We therefore decided to simply allow all such uses, and rather than regard this as a situation where the range of the property is ambiguous, think rather that the users of LIO are, by their use of this property on thousands of images, creating a category, which might be called the class of Settings.

We have no way to exactly characterize this class in a formal ontology, and it does not occur in any published ePedia, but we can say that it has a large overlap with what one might call spatiotemporal envelopes, and includes a wide range of things, such as *twilight* and *dusk*, which are hard to classify. It need not be shown or depicted in the image itself (although in some cases it can be, as in a room or a park or someplace one might be ‘in’ when they make the image) And this is enough to make it a useful and intuitive element of the LIO vocabulary. The key point is that users find it natural to use, and they use it consistently; which we take as evidence that it corresponds to a widely shared concept.

2.4 Foreground and Two Backgrounds

The notions of foreground and background are of course familiar, and we provided relations *hasInForeground* and *hasInBackground* to allow users to talk of objects in the foreground or background. But we quickly discovered that there are two additional notions of ‘background’ that could be utilized with a fair amount of ease once an annotator had learned they could use the terms. First, there is the flat 2-D background field of a rendered image versus the 3-D background of a depicted scene, the stuff ‘behind’ the main object depicted in an image. We needed the former in order to state precise search criteria for works of art (a painting by Keith Haring with a white background) when most ‘backgrounds’ were distant areas in photographic images. In this case, a real (but unforeseen) ambiguity had to be bounded by providing distinct IRIs for two subtly different but visually important senses of an English word; even though the meanings of our current relations *lio:hasPictoriaBackground*, *lio:hasDepictedBackground* are (of course) themselves still somewhat ambiguous, we find that when used, they are used consistently, perhaps because the very presence of the distinction in the small vocabulary draws user’s attention to fact that they can make this distinction explicit. *HasPictorialBackground* can refer to any kind of pattern or texture (such as ‘checkerboard’ or ‘swirls’) while *hasDepictedBackground* might be something like ‘sand’ behind a crab. These distinctions are valuable if perhaps one is searching for a pelican completely surrounded by water as opposed to sky (See Figure 2.)

Figure 2. Pelicans with Water as Depicted Background

My Images (2520) semantic search

Your search

- With property Image **lio:depicts** [delete](#) [change property](#) [choose property](#)

Pelican - Pelicans are a genus of large water birds that makes up the family Pelecanidae. ...

[Lookup more...](#)

- With property Image **lio:hasDepictedBackground** [delete](#) [change property](#)

Water -

Water - Water (chemical formula: H2O) is a transparent fluid which forms the world's str...

water - chemical compound formed of molecules composed of two atoms of hydrogen and one ...

Body of water - A body of water or waterbody (often spelled water body) is any significant accum...

Sunlight on Water - A visual phenomenon often used in imagery where sunlight is reflected from a liq...

water (inorganic material) - A liquid made up of molecules of hydrogen and oxygen (H2O). When pure, it is col...

pond (water) - Relatively small bodies of water, usually surrounded on all sides by land.

under-water - A photo taken under water

lake (body of water) - Bodies of fresh or salt water surrounded by land.

[Lookup more...](#)

[Refine](#)

Semantic Matches [\(see plain text matches\)](#)

Image **lio:hasDepictedBackground** (20 matches) (1 to 20)

						
Blue, Yellow, and Bird	Bird In Flight	testimage01_C.jpg	1941368	0491 What the ...	Bird In Flight	Bird In Flight
						
Bird In Flight	Bird In Flight	Bird In Flight	About to Dive for Dinner	Bird In Flight	Bird and Water	Bird In Flight

2.5 Looking Like

Early in the LIO project we had a photograph of clouds and sunlight that some people said, depicted an ‘angel’. This gave rise to a debate. Did it depict an angel? Some said, obviously yes, since this was the whole point of the photograph. Some said, obviously no, since there are is no actual angel to be depicted, only clouds, sky and sunlight. Some said angels did not exist and others emphatically said that medieval religious art was full of depictions of angels. We resolved this contradiction by introducing the notion of one thing looking like another. The image *depicts* clouds, but *lio:looksLike* an angel. Fortunately, the argument about the *existence* of angels was resolved by <http://dbpedia.org/resource/Angel>.

What seemed at first to be a simple conceptual hack for a limited class of ‘illusions’ turned out to be immediately useful in a large variety of descriptive situations, and *lio:looksLike* is now easily used by annotators. When reading plain-text descriptions of images, we often see phrases “looks like a bird” or “looks like a face”. One might call this the visual equivalent of metaphoric language.

2.6 Conveys

Early in the project, marking up a blue period Picasso, someone wanted to say that it *shows* sadness. This produced immediate objections, such as where exactly was this shown in the image? Again, the resulting disagreement/discussion led to a new conceptual distinction, and the addition of *lio:conveys* to the vocabulary, intended to refer to emotions and moods that images do not in any sense show or depict. But just as with *lio:hasSetting*, *lio:conveys* gets used in ways we had not initially anticipated and yet make perfect sense. So, for example an image can *lio:conveys* emptiness, friendship or even a ‘party’ (as in a ‘party atmosphere’) and so we realized that annotators might also use it to refer to an idea. So far there seems to be no reason to further delineate this concept. By using the search interface in the system, we have

seen that the use of the ‘conveys’ property is basically never used to ‘convey’ something ‘depicted’.

2.7 Artistic considerations

The final two properties added to LIO were: `hasArtisticElement` and `lio:usesPictorially`. Throughout the design of the interface and the design of the ontology, we were using teams of annotators who would annotate small groups of images and then run multiple types of searches over the freshly created data so that we could examine the usage of both the vocabulary and the similarity in the object terms chosen from (mostly) DBpedia and Yago. During this testing phase, we began to observe patterns in the usage of the terms and realized that the teams were naturally aligning themselves to their own conventions on edge cases. Take, for example a photo of a guitar in which the guitar takes up a large area of the image, but is still clearly not the only thing in the image. Does the image ‘depict’ a guitar or ‘show’ a guitar? In the beginning we might have seen annotators choose ‘depicts’ until they ran a search and observed that there was clearly a visible difference in the search results and thus, they aligned themselves using the images themselves as guides. While these testing sessions were occurring, we additionally discovered two other properties that were eagerly embraced, one was for the notion of an Artistic Element. We began searching for terms like: spirals, movement, shadow and blur; words the annotators had been using, and realized that a category of general concepts related to classical art theory design elements: line, form, shape and color attributes could be useful. Once the property existed, annotators quickly began using it to describe things like ‘vanishing points’ and ‘symmetry’.

The last property added was one called: *lio:UsesPictorially*. In practical use, while annotators were “triple-tagging” images and then testing their use of the properties, another distinction between objects was observed in the search results that could not be handled by any of the other properties. The distinction was a subtle difference in the way one would say they ‘normally’ see something real in the world depicted and then those items that seem to be visually represented in an unusual way – perhaps an artistic sense, or maybe just photographed in an unconventional way. This distinction most often includes artistic renderings of objects in the form of paintings or illustrations, but it often goes beyond that into abstract photos of everyday objects or unusual angles. Almost every new annotator learns this property very quickly and though there can be ambiguous edge cases here as well, this property has seemed to bear a great deal of utility in search results. Figure 1 illustrates the distinctions between depicts, shows and usesPictorially in images of guitars

3. Describing what is depicted

While working on the design of LIO, we were also working on the design of the system itself. The core of the ImageSnippets interface is an editor window called the ‘triple editor’ in which most of the user activity occurs. Over the years we have experimented with several design options and consider this part of the system a work in progress.

Our goal has been to make the process of finding, and sometimes of coining, concept IRIs as quick and as painless as possible and at the moment, we have multiple ways of accomplishing the creation of the triples including ways to select multiple images at one time to annotate all of them with the same triple(s) simultaneously as well as entity extraction from free text input provided by subject-matter experts (SMEs) and direct image-to-word suggestions made by Clarifai and CloudSight, as shown in figure 3.

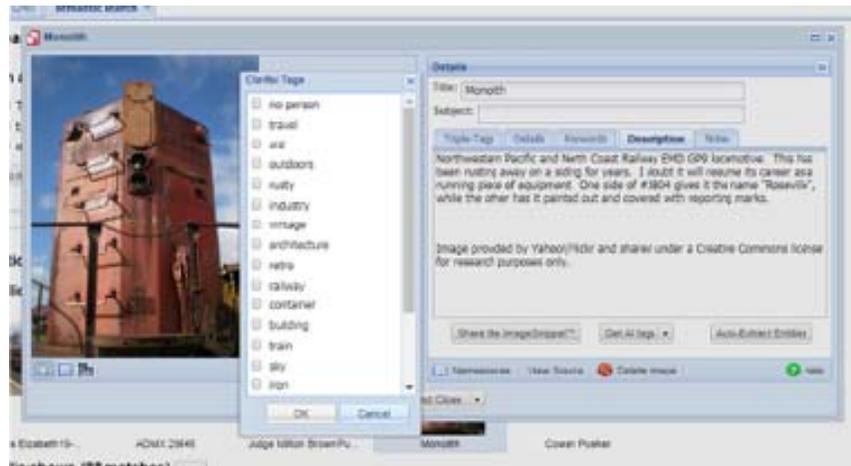


Figure 3: Clarifai (AI generated keyword suggestions) and the free text description by the SME

The basic creation of a triple involves the user typing in a word or phrase. At this point, we have a large number of concepts that annotators have previously found in all of the public corpora we have used to date. These concepts are auto-completed and presented to the user who can hover over the concepts to make sure they are disambiguating the term correctly. This auto-complete function in itself aids in the alignment of ambiguous results, since it is natural for a user to select an entity that has previously been chosen if it matches their intended sense of the concept they are describing. However, if it is a new concept – not previously used in the system, or if the user chooses – they can go to a full lookup and scan through a look-up window that shows all of the possible entities that might match the term the user has typed to find a correct definition that matches the sense of the word they had in mind. This can often be the most time consuming stage of creating the triples and one of the most challenging aspects of the design of the system.

It was our original intent to use just one public dataset to look up terms; however, we quickly found that not anyone dataset by itself contained enough range of concepts to allow for the expressiveness of real annotation. So, while the system allows both custom datasets to be used and the creation of new datasets; the default look-up

displays results from DBpedia, Yago, Art and Architecture Thesaurus and Wikidata. This, in and of itself can lead to duplications in triples across different datasets, but also occasionally forces annotators to make arbitrary choices among needlessly fine distinctions of meaning. The word ‘curve’ for example yields two senses in Yago: a curved section of a road or railway (the object itself) or the property of something (such as a roadway) being curved. This excess of conceptual distinctions based on fine philosophical or grammatical considerations or simple due to duplication is a source of difficulty for our project. Still, we cannot ‘not’ show all results, because in some cases – that fine grained distinction *is* necessary.

While the various corpora have many *owl:sameAs* links identifying similar concepts, these are not useful in practice. They are unreliable; but in any case, reasoning with equality is expensive and slow. What is needed is to have a single source of canonical names for concepts at just the right level of ambiguity and for now, we have made some headway in resolving this sometimes immense ‘embarrassment of concepts’ by hand over the course of hundreds of person-hours to build the cached concepts found by auto-completion. We also use some programmatic tricks that organize the results in the look-up such that the most likely matches will be at the top of each dataset. In any case, over time, power annotators quickly learn the fastest paths to disambiguation and this leads to increasingly more fluid look-up overall.

4. Visual Ontology Building

The development of the system has also given us multiple opportunities for experimentation in ontology development. First, we added the ability for the annotators to build more complex descriptions by chaining triples together by use of a ‘which’ and ‘and’ statements. In this way, more skilled annotators can not only describe the objects seen in the image, but they are also creating a slowly growing corpus of useful concepts - mostly RDF properties which seem to allow quite rich expressiveness within the confines of the simple RDF triple graph syntax. As images can depict just about anything, even mythological or illusory things, this process often pushes the edge of the expressive abilities of the published vocabularies, requiring power users to create new properties as well as new nominal concepts. Figure 4 illustrates an example of triples chained together with new ‘properties’ which are being accumulated in the system for further analysis. Often these are concepts arising in specialized domains, such as *cookingMethod* and *FoodStage* used to describe images illustrating recipes, *isWearing* used to talk about people in an image, and *hasVIN* to identify particular cars. But some are of more general utility, such as *isPartOf*, *isMadeOf*, and *hasColor*. Some of these have become extended in use, just as in the case of *lio:hasSetting*. One in particular, *hasCondition*, was introduced for describing stages in a process, such as a building *under construction* or a car engine *being assembled*, but its use has extended to things like *a curved road*, made up from the available concepts of *dbpedia:Road* and *yago:curve*. It can be used now to indicate a complex concept in which one descriptor modifies another. In many ways this seems to capture the adjectival construction in English, while staying within the simple

triple-graph syntax of RDF. This process of Darwinian selection (by frequency of re-use) and conceptual generalization is evolving a general-purpose conceptual language for describing things, which builds on and extends the basic semantic corpora; and is of course itself, uniquely, documented and illustrated by the images themselves.

Another style of usage for ontology development has also been found especially productive in which subject-matter experts (SMEs) provide image descriptions in free text, which is then processed by trained ImageSnippets power annotators who create the actual triple, using multiple techniques for more rapid triple building including the use of an auto-entity extractor from DBpedia over the free text which users can then automatically add to a list of pre-disambiguated concepts for attaching to LIO or their own user-created properties.

In cases like this, the concepts used by the SME are often more specialized, reflecting their expertise, than those found in the conceptual corpus. For example, a vehicle may be identified as a rare 1953 Fletcher prototype (one of only two still in existence) rather than simply a military vehicle, which is the best suggestion made by



Figure 4. Use of chained triples in the Triple Editor

Clarifai. In these cases, the annotator can generate new IRI's for such terms, while noting and recording the subclass or instance relationships between these and the

existing terms in the public corpus. In this way, a specialized ontological vocabulary can be created as a byproduct of the image annotation process.

5. Conclusion

In conclusion, our long-term image annotation project, predominantly a labor of love by the first author, has had a number of goals, the first and foremost of which has been the construction of a linked-data, image annotation ‘sandbox’ for extended research and development in the areas of knowledge capture and representation using images as stimulus. Secondary goals involve the more practical use of sharing, managing and publishing the images and their metadata as well as an interest in using the system for the curation and preservation of historically significant digital image assets, particularly those from niche or at-risk domains with extremely deep and precise metadata. Much of our work has involved highly subjective and ambiguous distinctions, trial and error, testing and re-testing while simultaneously making interface changes to improve usability and the training of annotators to work in unfamiliar mental territory. Our methodology, in favor of intuitive usability has embraced a process of creating boundaries around ambiguity eschewing rigid definitions. Through a process of debating and resolving disagreements by splitting concepts, we attempt to identify useful conceptual generalizations as they emerge and adjust the boundaries of natural meanings. This is work in progress, but we are optimistic.

6. References

- Eskridge, T., Hoffman, R., Hayes, P. and Warren, M. 2006 Formalizing the informal: a Confluence of Concept Mapping and the Semantic Web. *Proc. Second International Conference on Concept Mapping*, A J Cañas & J. Novak, eds. Costa Rica, 2006 □
- Warren, M., and Hayes, P. 2007 Artspeak: The Contemporary Artist Meets the Semantic Web. Creating Formal Semantic Web Ontologies from the Language of Artists *Electronic Techtonics - Thinking at the Interface*; HASTAC (Humanities, Arts, Science, and Technology Advanced Collaboratory) - Duke University, 2007 □ □
- Brickley, D. and Miller, L. 2010 FOAF Vocabulary Specification 0.97 <http://xmlns.com/foaf/spec/20100101.html>
- Warren, M, 2016 A Visual Guide to the ImageSnippets Properties. <http://www.imagesnippets.com/ArtSpeak/help/properties.html>

